IN THE CLAIMS:

1. (Currently amended) For use in a wide-issue processor, a mechanism for identifying and tracking conditional instructions, comprising:

a conditional execution block state machine that tags and generates contemporaneous link pointers for instructions located in a conditional execution block, wherein said link pointers mark at least the beginning and end of a conditional execution block of instructions; and

conditional link pointer register sets, wherein each of said sets corresponds to a stage of a pipeline of said processor, that contain and cause said link pointers to move through each of said sets as said instructions associated with said link pointers and located in said conditional execution block move through each of said corresponding stages.

2. (Previously Presented) The mechanism as recited in Claim 1 further comprising a conditional execution marking queue, associated with said conditional execution block state machine, that contains ones of said link pointers prior to storage in said sets.

3. (Original) The mechanism as recited in Claim 2 wherein said conditional execution marking queue is a five-bit, six-entry queue and comprises a reordering multiplexer.

4. (Original) The mechanism as recited in Claim 1 further comprising a conditional execution attribute register, associated with a group stage of said pipeline, that contains an attribute associated with one of said eonditional instructions.

5. (Cancelled)

6. (Original) The mechanism as reeited in Claim 4 further comprising a conditional execution attribute queue that contains attributes read from said conditional execution attribute register.

7. (Original) The mechanism as recited in Claim 6 wherein said conditional execution attribute queue is of variable depth and comprises a selecting multiplexer.

8. (Currently amended) For use in a wide-issue processor, a method of identifying and tracking conditional instructions, comprising:

generating tags and contemporaneous link pointers for instructions located in a conditional execution block; wherein said generating includes marking at least the beginning and end of a conditional execution bloek of instructions; and

causing said link pointers to move through conditional link pointer register sets, wherein each of said sets corresponds to a stage of a pipeline of said processor, as said instructions associated with said link pointers and located in said conditional execution block move through each of said corresponding stages.

9. (Previously Presented) The method as reeited in Claim 8 further eomprising containing ones of said link pointers in a conditional execution marking queue prior to storage in said sets.

10. (Original) The method as recited in Claim 9 wherein said conditional execution marking queue is a five-bit, six-entry queue and comprises a reordering multiplexer.

11. (Original) The method as recited in Claim 8 further comprising containing an attribute associated with one of said conditional instructions in a conditional execution attribute register.

12. (Cancelled)

13. (Original) The method as recited in Claim 11 further comprising containing attributes read from said conditional execution attribute register in a conditional execution attribute queue.

14. (Original) The method as recited in Claim 13 wherein said conditional execution attribute queue is of variable depth and comprises a selecting multiplexer.

15. (Currently amended) A wide-issue digital signal processor (DSP), comprising:

a pipeline having stages capable of executing instructions conditionally;

a wide-issue instruction issue unit;

a conditional execution block state machine, associated with said instruction issue unit, that tags and generates contemporaneous link pointers for instructions located in a eonditional execution block, wherein said link pointers mark at least the beginning and end of a conditional execution block of instructions; and

conditional link pointer register sets, wherein each of said sets corresponds to a stage of said pipeline, that contain and cause said link pointers to move through each of said sets as said instructions associated with said link pointers and located in said conditional execution block move through each of said corresponding stages.

16. (Previously Presented) The DSP as recited in Claim 15 further comprising a conditional execution marking queue, assoeiated with said conditional execution bloek state machine, that contains ones of said link pointers prior to storage in said sets.

17. (Original) The DSP as recited in Claim 16 wherein said conditional execution marking queue is a five-bit, six-entry queue and eomprises a reordering multiplexer.

18. (Original) The DSP as recited in Claim 15 further comprising a conditional exeeution attribute register, associated with a group stage of said pipeline, that contains an attribute associated with one of said conditional instruetions.

19. (Cancelled)

20. (Original) The DSP as reeited in Claim 18 further comprising a conditional execution attribute queue that contains attributes read from said conditional exeeution attribute register.

21. (Original) The DSP as recited in Claim 20 wherein said conditional execution attribute queue is of variable depth and comprises a selecting multiplexer.

22. (New) The mechanism as recited in claim 1, wherein said link pointers move through each register of each register set as at least one instruction move through each of said corresponding stages.

23. (New) The method of claim 8, further comprising:

moving link pointers through each register of each register set as at least one instruction move through each of said corresponding stages.